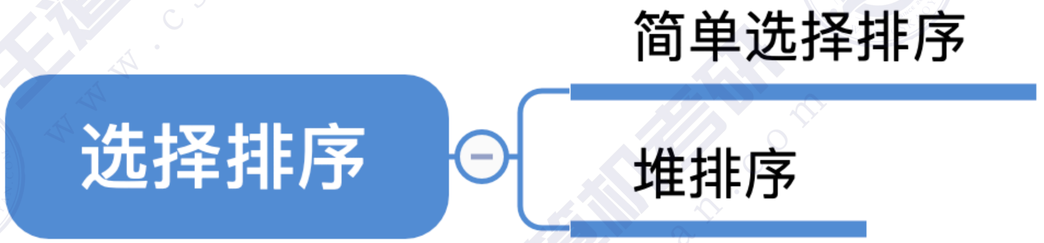
本节内容

简单选择排序

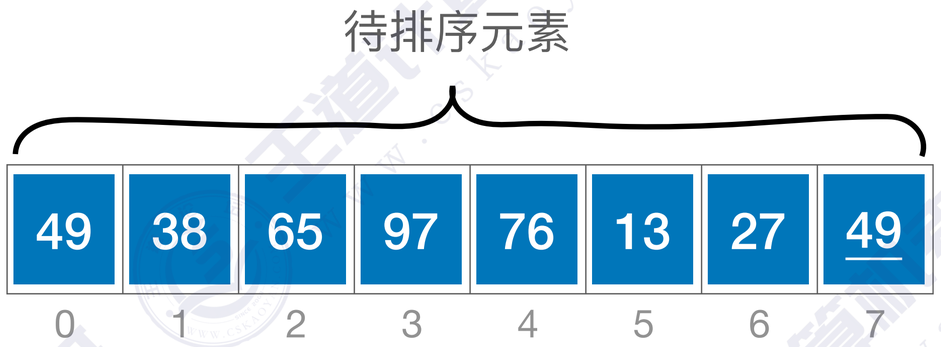
知识总览



选择排序：每一趟在待排序元素中选取关键字最小（或最大）的元素加入有序子序列

简单选择排序

每一趟在待排序元素中选取关键字最小的元素加入有序子序列

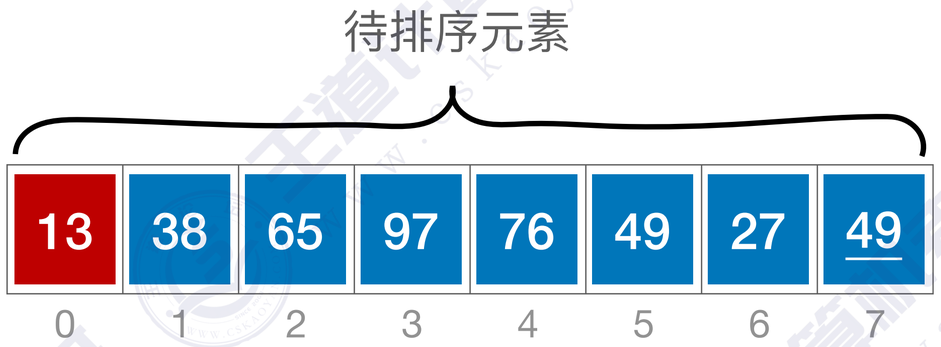


简单选择排序

每一趟在待排序元素中选取关键字最小的元素加入有序子序列

简单选择排序

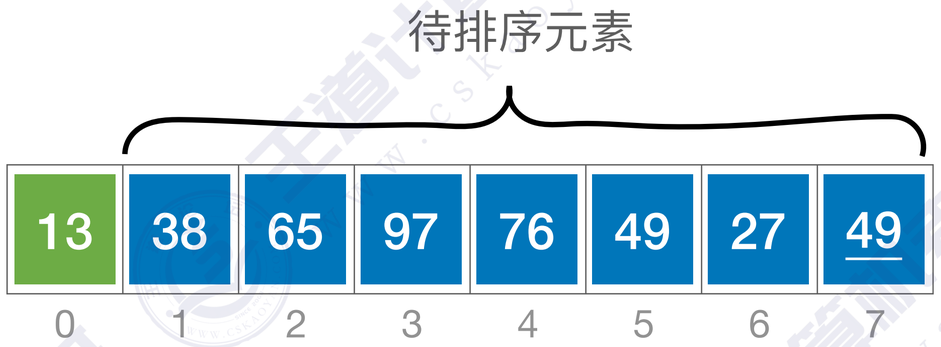
每一趟在待排序元素中选取关键字最小的元素加入有序子序列



简单选择排序

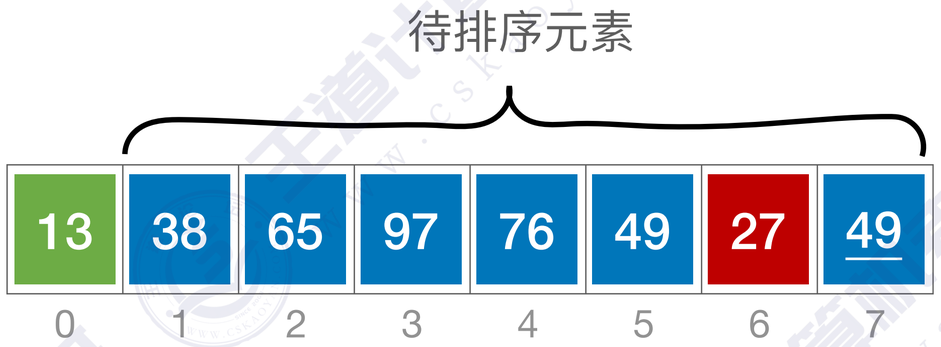
每一趟在待排序元素中选取关键字最小的元素加入有序子序列

第1趟排序结束：



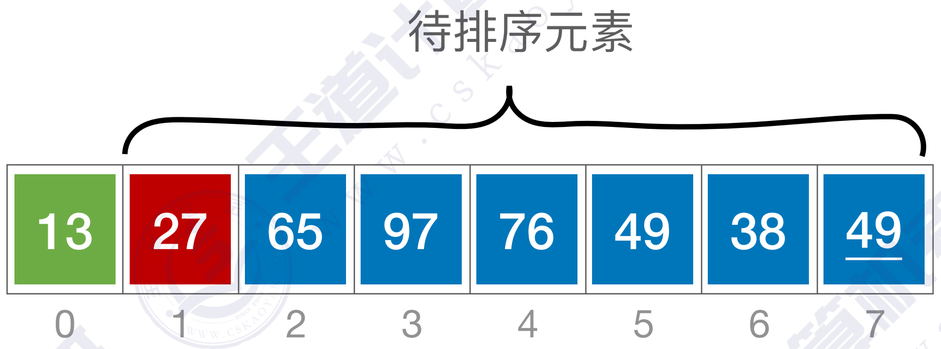
简单选择排序

每一趟在待排序元素中选取关键字最小的元素加入有序子序列



简单选择排序

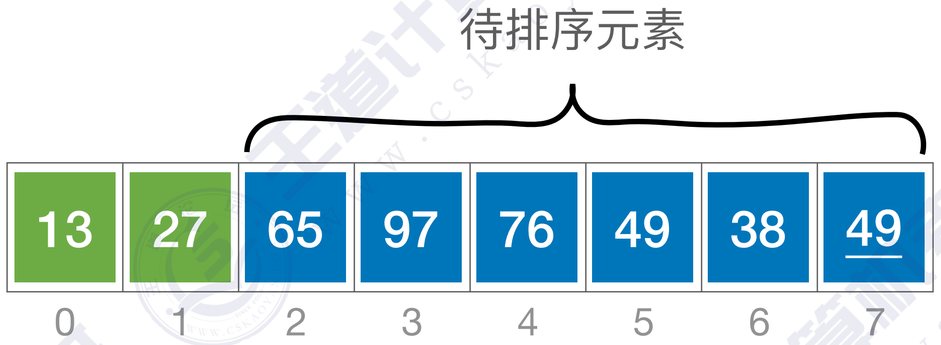
每一趟在待排序元素中选取关键字最小的元素加入有序子序列



简单选择排序

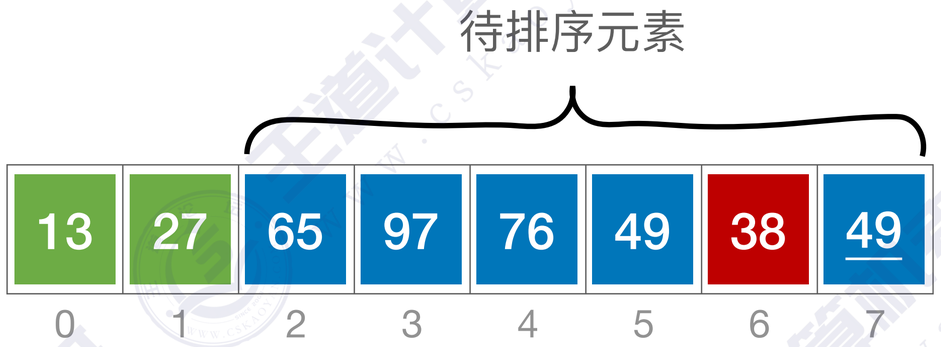
每一趟在待排序元素中选取关键字最小的元素加入有序子序列

第2趟排序结束：



简单选择排序

每一趟在待排序元素中选取关键字最小的元素加入有序子序列



简单选择排序

每一趟在待排序元素中选取关键字最小的元素加入有序子序列

简单选择排序

每一趟在待排序元素中选取关键字最小的元素加入有序子序列

第3趟排序结束：

简单选择排序

每一趟在待排序元素中选取关键字最小的元素加入有序子序列

简单选择排序

每一趟在待排序元素中选取关键字最小的元素加入有序子序列

简单选择排序

每一趟在待排序元素中选取关键字最小的元素加入有序子序列

第4趟排序结束：

简单选择排序

每一趟在待排序元素中选取关键字最小的元素加入有序子序列

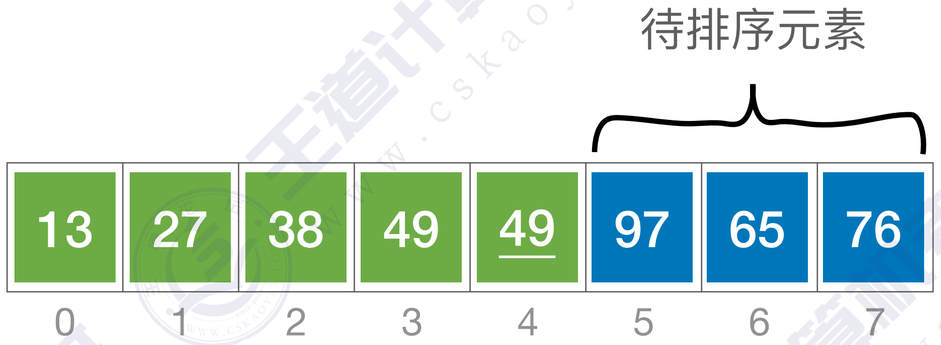
简单选择排序

每一趟在待排序元素中选取关键字最小的元素加入有序子序列

简单选择排序

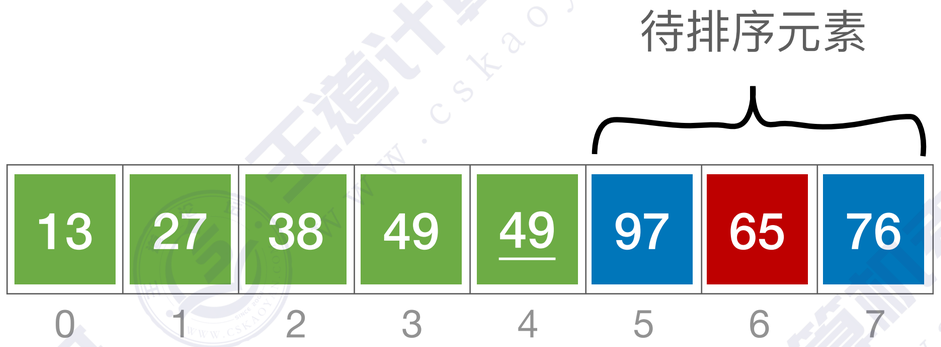
每一趟在待排序元素中选取关键字最小的元素加入有序子序列

第5趟排序结束：



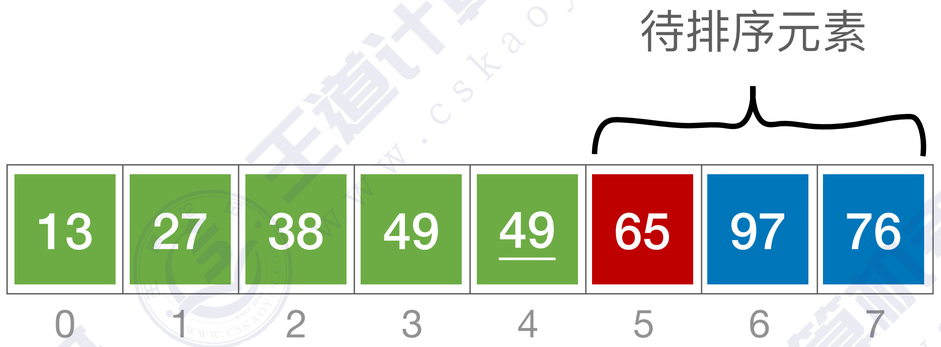
简单选择排序

每一趟在待排序元素中选取关键字最小的元素加入有序子序列



简单选择排序

每一趟在待排序元素中选取关键字最小的元素加入有序子序列



简单选择排序

每一趟在待排序元素中选取关键字最小的元素加入有序子序列

第6趟排序结束：

简单选择排序

每一趟在待排序元素中选取关键字最小的元素加入有序子序列

简单选择排序

每一趟在待排序元素中选取关键字最小的元素加入有序子序列

简单选择排序

每一趟在待排序元素中选取关键字最小的元素加入有序子序列

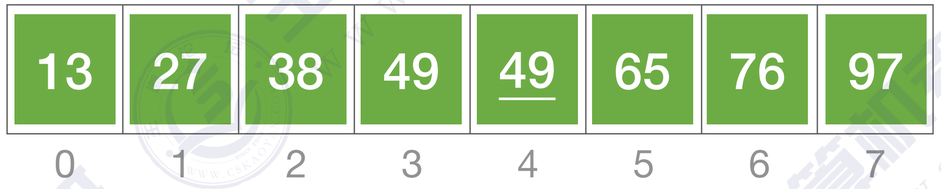
第7趟排序结束：

简单选择排序

每一趟在待排序元素中选取关键字最小的元素加入有序子序列

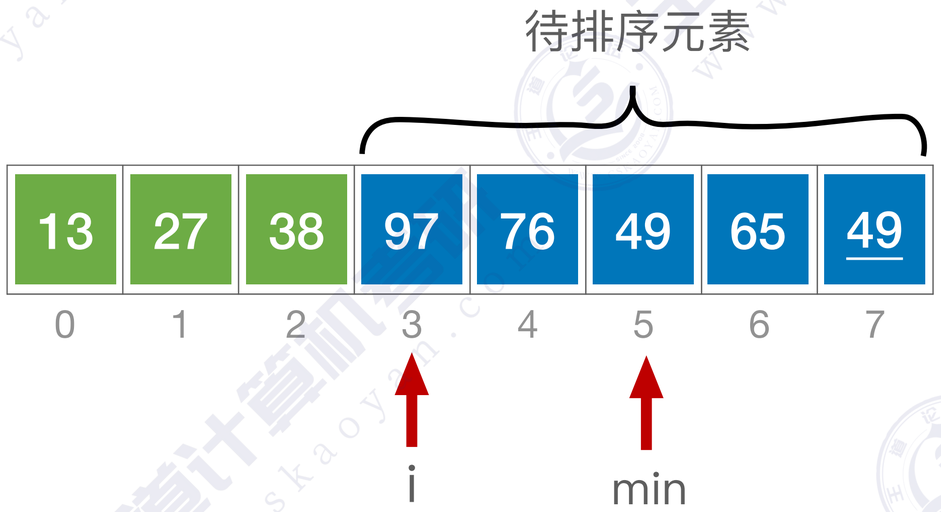
n个元素的简单选择排序需要 n-1 趟处理

第7趟排序结束：



算法实现

```
//简单选择排序
void SelectSort(int A[],int n){
    for(int i=0;i<n-1;i++){                  //一共进行n-1趟
        int min=i;                           //记录最小元素位置
        for(int j=i+1;j<n;j++)               //在A[i…n-1]中选择最小的元素
            if(A[j]<A[min]) min=j;           //更新最小元素位置
        if(min!=i) swap(A[i],A[min]);        //封装的swap()函数共移动元素3次
    }
}
```

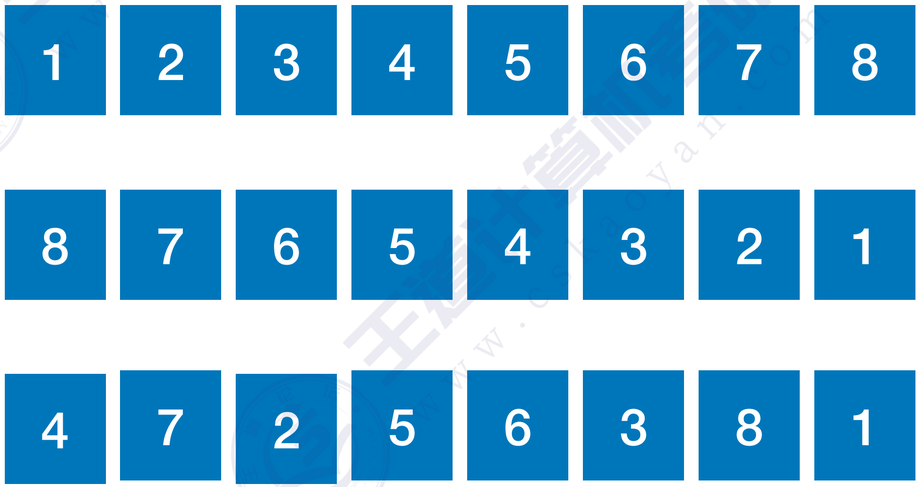



```
//交换
void swap(int &a, int &b){
    int temp = a;
    a = b;
    b = temp;
}
```

算法性能分析

空间复杂度：O(1)

时间复杂度=$O(n^2)$



无论有序、逆序、还是乱序，一定需要 n–1 趟处理

总共需要对比关键字 (n–1)+(n–2)+…+1 = $\frac{n(n-1)}{2}$ 次

元素交换次数 < n–1

算法性能分析

第1趟排序结束：

第2趟排序结束：

稳定性：不稳定

适用性：既可以用于顺序表，也可用于链表

知识回顾与重要考点